

Re: How to determine whether any one is logged on (VB6, Win2K and later)

Source:

<http://www.tech-archive.net/Archive/VB/microsoft.public.vb.general.discussion/2007-12/msg00622.html>

- *From:* akaTwoSheds <akaTwoSheds@xxxxxxxx>
 - *Date:* Mon, 10 Dec 2007 03:22:19 -0800 (PST)
-

On Dec 9, 3:51 pm, akaTwoSheds <akaTwoSh...@xxxxxxxx> wrote:

Greetings all.

Hopefully I can explain this in a semi-comprehensible fashion. We have a VB6 application and several of our clients have recently asked if it can be set up to run as a service. It's only being run on Win2K or later systems so at least we don't have to worry about 9x compatibility. There's a swell little freeware program called XYNTService (<http://www.codeproject.com/KB/system/xyntservice.aspx?fid=1239&noise=3&df=90&mpp=25&sort=Position&view=Quick&fr=276>) which itself runs as a service and then fires the programs specified, so life is good (I should add that we are working on a slightly more up to date version of the application but it's no where near ready and we're facing time pressure).

Using XYNTService everything works great, the application loads up when Windows starts, continues running as people log in and log out and is generally perfect. The only problem is that the user interface is activated by clicking an icon in the system tray. The code which creates that tray icon is executed once during the program start up, which, when being run by XYNTService, happens long before anyone has logged in and therefore before the system tray exists.

In otherwords, I can't make it appear when I want it to.

It's a simple matter to run a timer every 30 or so seconds to create the icon, but it gets complex when the icon has been successfully added. Naturally I don't want to keep hitting that timer once the job is done, but so far I've not been able to find a way to tell that the tray icon has been created.

To my mind there are two things I can check for. The existence of the tray icon, or the presence of a user logged into the PC directly.

I'm using Mark Mocoski's SysTray Module to manage the tray icon. Here's the relevant code (I've eliminated the many things I've tried

Re: How to determine whether any one is logged on (VB6, Win2K and later)

so you can see the actual code rather than my ham handed attempts):

```
Public Declare Function ShellExecute Lib "shell32" Alias  
"ShellExecuteA" & _  
(ByVal hWnd As Long, & _  
ByVal lpOperation As String, & _  
ByVal lpFile As String, & _  
ByVal lpParameters As String, & _  
ByVal lpDirectory As String, & _  
ByVal nShowCmd As Long) As Long
```

```
Public Declare Function Shell_NotifyIcon Lib "shell32" Alias  
"Shell_NotifyIconA" & _  
(ByVal dwMessage As Long, & _  
pnid As NOTIFYICONDATA) As Boolean
```

```
Public Declare Function SetForegroundWindow Lib "user32" & _  
(ByVal hWnd As Long) As Long
```

```
Public Declare Function SetWindowPos Lib "user32" & _  
(ByVal hWnd As Long, & _  
ByVal hWndInsertAfter As Long, & _  
ByVal x As Long, & _  
ByVal Y As Long, & _  
ByVal cx As Long, & _  
ByVal cy As Long, & _  
ByVal wFlags As Long) As Long
```

```
Public Type NOTIFYICONDATA  
cbSize As Long  
hWnd As Long  
uID As Long  
uFlags As Long  
uCallbackMessage As Long  
hIcon As Long  
szTip As String * 128  
dwState As Long  
dwStateMask As Long  
szInfo As String * 256  
uTimeout As Long  
szInfoTitle As String * 64  
dwInfoFlags As Long  
End Type
```

```
Public multiTip As Boolean 'In shell32.dll  
ver 5, exploit "bug" and allow multi balloon tips  
Public blnClick As Boolean  
Public vbTray As NOTIFYICONDATA
```

Re: How to determine whether any one is logged on (VB6, Win2K and later)

```
Public Const SWP_NOMOVE As Long = &H2
Public Const SWP_NOSIZE As Long = &H1
Public Const flags As Long = SWP_NOMOVE Or
SWP_NOSIZE
Public Const WM_RBUTTONDOWN As Long = &H205
Public Const WM_RBUTTONDOWNCLK As Long = &H204
Public Const WM_LBUTTONDOWN As Long = &H202
Public Const WM_LBUTTONDOWNBLCLK As Long = &H203
Public Const WM_MOUSEMOVE As Long = &H200
Public Const NIM_ADD As Long = &H0
Public Const NIM_DELETE As Long = &H2
Public Const NIF_ICON As Long = &H2
Public Const NIF_MESSAGE As Long = &H1
Public Const NIM_MODIFY As Long = &H1
Public Const NIM_SETVERSION As Long = &H4
Public Const NIF_TIP As Long = &H4
Public Const NIF_INFO As Long = &H10
Public Const NIS_HIDDEN As Long = &H1
Public Const NIS_SHAREDICON As Long = &H2
Public Const NIIF_NONE As Long = &H0
Public Const NIIF_WARNING As Long = &H2
Public Const NIIF_ERROR As Long = &H3
Public Const NIIF_INFO As Long = &H1
Public Const NIIF_GUID As Long = &H4 'Ver 6 of
Shell32.dll (WinXP SP2) only
Public Const HWND_NOTOPMOST As Long = -2
Public Const HWND_TOPMOST As Long = -1
Public Const NOTIFYICON_VERSION As Long = 3

Public Sub SystrayOn(frm As Form, IconTooltipText As String)

    With vbTray
        .cbSize = Len(vbTray)
        .hWnd = frm.hWnd
        .uID = vbNull
        .uFlags = NIF_ICON Or NIF_TIP Or NIF_MESSAGE
        .uCallbackMessage = WM_MOUSEMOVE
        .szTip = Trim(IconTooltipText) & vbNullChar
        .hIcon = frm.Icon
    End With
    Call Shell_NotifyIcon(NIM_ADD, vbTray)
    App.TaskVisible = False

End Sub

Form_Load

Call SysTray(frmMain, "Tray Icon")

End Sub
-----
```

Re: How to determine whether any one is logged on (VB6, Win2K and later)

I've tried adding a flag to indicate that the icon has been created, but the code executes without an error even though there's no system tray for it to put the icon in, so as far as it's concerned all is swell.

I've also tried setting and then checking the szTip, szInfo, and szInfoTitle attributes hoping that when the tray became active one of more of these would change, but no such luck.

Next I tried the GetUserName API but of course that returns the user name of the service which is running the application, not the name of the user who's just logged into Windows. On the low tech end I also tried checking the system environment variables but got the same results.

Can anyone think of a workable band aid for this? Long term the plan is to separate the UI from the main portion of the program so that the program can be run as a true service, and the interface can only be called up as needed but we're a far ways off from that point.

Thanks for any ideas suggestion or at this point fervent prayers :)

Thanks for all the feedback, and trust me, splitting the app into two distinct programs, one encapsulating the UI and another which would be the service is very high on the list of things to do. The trouble is that there's literally no way on Earth to accomplish that before the necessary date, hence this Rube Goldberg band aid to buy us the necessary time to complete the redesign.

As to why it has to be a service, mostly because a new client who more than quadruples our business says so :) . Actually they do have a good reason for it. The program was specifically designed to be run only when a user was logged into the workstation, either locally or with a domain ID. The new client wants to install it on servers and/or work stations, neither of which will likely have anyone logged in most of the time.

Unfortunately they also want to be able to see it before they go on Christmas holiday which starts on 12/21.

In playing around some I've had minor success with enumerating the active processes and checking for "Program Manager" in the list as an indicator that someone has logged in. So far it seems to be working but I've only tested it on a handful of PCs. If anyone has a better band aid I'd love to hear about it.

Thanks again.

.

Re: How to determine whether any one is logged on (VB6, Win2K and later)